UNIQUE ADDRESS SPACE AND METHOD
FOR A TRANSPORT NETWORK

RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application Serial No. 60/202,190, entitled INTERNET PROTOCOL TRANSPORT, filed May 5, 2000 which is hereby incorporated by reference.

TECHNICAL FIELD OF THE INVENTION

The present invention relates generally to the field of telecommunication networks, and more particularly to a unique address space and a method for a transport network.

BACKGROUND OF THE INVENTION

Telecommunication networks transport voice and data according to a variety of standards and using a variety of technologies. Circuit-switch networks such as plain old telephone service (POTS) utilize transmission paths dedicated to specific users for the duration of a call and employ continuous, fixed-bandwidth transmission. Packet-switch networks (PSNs) allow dynamic bandwidth, depending on the application, and can be divided into connectionless networks with no dedicated paths and connection-oriented networks with virtual circuits having dedicated bandwidth along a predetermined path. Because packet-switched networks allow traffic from multiple users to share communication links, these networks utilize available bandwidth more efficiently than circuit-switched networks.

Internet protocol (IP) networks are connectionless packet-switched networks. IP networks transport information by breaking up bitstreams into addressable digital packets. Each IP packet includes source and destination addresses and can take any available route between the source and the destination. The IP packets are transmitted independently and then reassembled in the correct sequence at the destination.

IP networks have limited address space. As a result, the interconnection of discrete networks can cause conflicts between the native address spaces of the networks. Readdressing networks to overcome conflicts is time consuming and expensive.

SUMMARY OF THE INVENTION

The present invention provides a unique address space and method for a transport network that substantially eliminate or reduce the problems and disadvantages associated with previous systems and methods. In a particular embodiment, the transport network utilizes an internal address space that is reserved and non-forwardable in external Internet protocol (IP) networks and translates between the internal address space and the external IP address space to prevent address conflicts between the networks and to reduce needed IP addresses.

In accordance with one embodiment of the present invention, a method and system for routing an externally generated message in a network includes receiving at an ingress port of a network a message from an external network including Internet protocol (IP) source and destination addresses and message data. The IP source and destination addresses are translated to internal addresses that are non-forwardable in the external network. The message data is routed in the network based on the internal addresses.

Technical advantages of one or more embodiments of the present invention include providing an improved transport network. In particular, the transport network utilizes an internal address space that is unusable by external IP networks for traffic routing. As a result, address conflicts between the networks are eliminated and necessary IP addresses to communicate with a transport network may be reduced.

Another technical advantage of one or more embodiments of the present invention includes providing improved security for a transport network. In particular, the address space of the transport network is isolated from the external network. The internal address is known only to ports of the transport network, and thus hidden to external elements.

Other technical advantages of the present invention will be readily apparent to one skilled in the art from the following figures, description, and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and its advantages, reference is now made to the following description taken in conjunction with the accompanying drawings, wherein like reference numerals represent like parts, in which:

FIGURE 1 is a block diagram illustrating a transport network in accordance with one embodiment of the present invention;

FIGURE 2 is a block diagram illustrating an external representation for the transport router of FIGURE 1 in accordance with one embodiment of the present invention;

FIGURE 3 is a block diagram illustrating details of the Internet protocol transport (IPT) node of FIGURE 1 in accordance with one embodiment of the present invention;

FIGURE 4 is a block diagram illustrating details of the receiver-transmitter pair (RTP) of FIGURE 3 in accordance with one embodiment of the present invention;

FIGURE 5 is a block diagram illustrating details of the processing system of FIGURE 3 in accordance with one embodiment of the present invention;

FIGURE 6 is a block diagram illustrating distribution of functionality between processors in an exemplary network in accordance with one embodiment of the present invention;

FIGURE 7 is a block diagram illustrating details of the transport network layer one (IPTL1) architecture for the processing system of FIGURE 5 in accordance with one embodiment of the present invention;

FIGURE 8 is a block diagram illustrating details of the transport element layer two (IPTL2) architecture for the processing system of FIGURE 5 in accordance with one embodiment of the present invention;

FIGURE 9 is a flow diagram illustrating a method for provisioning an IPT network in accordance with one embodiment of the present invention;

FIGURE 10 is a flow diagram illustrating a method for defining a transport router in an IPT network in accordance with one embodiment of the present invention;

FIGURE 11 is a flow diagram illustrating a method for generating routing tables for a transport router in accordance with one embodiment of the present invention;

FIGURE 12 is a flow diagram illustrating a method for processing through traffic in a transport router in accordance with one embodiment of the present invention; and

FIGURE 13 is a flow diagram illustrating a method for routing messages in a transport network using a unique internal address space and translating between the internal address space and an external IP address space in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

FIGURE 1 illustrates a transport network 10 in accordance with one embodiment of the present invention. In this embodiment, the transport network 10 is an Internet protocol (IP) network for transporting IP and Multiple Protocol Label Switch (MPLS) packets. The transport network 10 may be any other packet-switched network operable to route, switch, and/or otherwise direct data packets based on network protocol addresses.

The transport network 10 is a private network connecting geographically distributed segments of an external network 12. The external network 12 includes one or more public and/or private networks such as the Internet, an intranet, and other suitable local area networks (LAN), wide area networks (WAN), and nodes. The external network 12 includes label switch and subtending routers 14, Ethernet switches 16, Frame Relay switches 18, management station 20 and other suitable routers, switches, and nodes operable to generate and/or transport traffic. The transport network 10 communicates with nodes of the external network 12 in the native protocol of the nodes to communicate traffic and control signaling between the networks 10 and 12.

Referring to FIGURE 1, the transport network 10 includes a plurality of Internet protocol transport (IPT) nodes 30 interconnected by communication links 32. The IPT nodes 30 each include a plurality of ports 34 accessible to the external network 12. As used herein, each means every one of at least a subset of the identified items. The communication links 32 are optical fiber or other suitable high-speed links. The high-speed links are operable to transport traffic at a rate of 5

Gb/s or greater. Preferably, the high-speed links 32 transport traffic at rates of 10 Gb/s or above.

As described in more detail below, the high-speed links 32 connect high speed interfaces of the IPT nodes 30 to form fast transport segments (FTS) through the transport network 10. Packets transferred via the FTSs incur very small buffering delay in the network as described in co-owned U.S. Patent Application entitled "*Method and System for Transporting Traffic in a Packet-Switched Network*", filed June 6, 2000. Packets carried through the ports 34 and between FTSs may incur queuing delay comparable to a normal IP switch.

To optimize bandwidth usage within the transport network 10, packets may be transmitted directly on the high-speed optical links 32 without synchronous optical network (SONET) framing and its associated overhead which imposes a penalty of three to five percent depending on the line rate. In one embodiment, a transport label is added to each packet to generate an internal packet that can be directly transmitted on the optical links 32. Details of the transport label are described in co-owned U.S. Patent Application entitled "*System and Method for Connectionless/Connection Oriented Signal Transport*", filed June 6, 2000. Using the transport label, both connection-oriented and connectionless traffic may be seamlessly transported across the transport network 10. Protection for connection oriented data flows may be provided as described in co-owned U.S. Patent Application entitled "*Method and System For Providing A Protection Path For Connection-Oriented Signals In A Telecommunications Network*", filed June 6, 2000. Protection for connectionless, packet transport, traffic flows may be provided as described in co-owned U.S. Patent Application "*Method and System For Providing A Protection Path For Connectionless Signals In A Telecommunications Network*", filed June 6, 2000.

To support voice, video, and other real-time or time-sensitive applications, the transport network 10 may provide class of service (CoS) capabilities. In one embodiment, all IP packets are mapped to one of three priority levels as they enter the transport network 10. In this embodiment, guaranteed traffic has reserved bandwidth and is guaranteed to be transported within a defined time delay. Control flow traffic is also reserved and guaranteed, but the network 10 does not guarantee delivery time delay. Best effort traffic does not have reserved bandwidth and delivery is not guaranteed by the network 10. By distinguishing and prioritizing traffic based on its type, including CoS, service level agreement (SLA) and/or other suitable indication of importance or delivery constraints. The transport network 10 is able to deliver time-sensitive traffic within tight time constraints by delaying and/or dropping best effort traffic and other low priority traffic.

In one embodiment, the transport network 10 utilizes a private internal addressing scheme to isolate the network 10 from customers and thus minimize or prevent conflicts with private and/or public networks connected to the transport network 10. This reduces the complexity of network management and preserves the topology of the existing routed network 12. In addition, transport network isolation enables value added services to be provided through the transport network 10.

When an independent addressing scheme is utilized for the transport network 10, egress traffic is converted from the external addressing scheme to the internal addressing scheme at ports 34 using standardized or extended network address translation (NAT). Similarly, egress traffic is converted from the internal addressing scheme back to the external addressing scheme at ports 34 using standard or extended NAT. In addition to the internal addresses, each IPT node 30, port 34 and other component of the transport network 10 visible to the external network 12 includes a globally unique IP address. These addresses are used for external management of the transport network 10.

The transport network 10 provides a flexible topology in which sets of ports 34 may be grouped in any suitable way and each treated as a single entity capable of independently interacting with external nodes. Thus, the transport network 10 is externally represented as sets of port groups 50 with internally managed connectivity. Provisioning of port groups 50 in the transport network 10 is unconstrained with mesh and partial-mesh topologies supported.

The port groups 50 are each a set of ports 34 with similar routing properties. In particular, a port group 50 is a set of ports 34 configured to provide multipoint-to-multipoint or at least point-to-multipoint connectivity between one another which allows point-to-multipoint connectivity between external elements. Accordingly, traffic received by a port group 50 can be routed directly from an ingress port 34 to a plurality of egress ports 34 without channelization in the transport network 10.

Port groups 50 may be provisioned as simple port groups or as composite port groups. In the simple port group configuration, each port 34 only belongs to a single port group 50. Private addresses can be supported inside the simple port group configuration. A composite port group includes ports 34 which have membership in multiple port groups 50. In the composite port group case, private IP addressing is not supported.

The port groups 50 each define a transport element 52 with geographically distributed ports 34. Each transport element 52 is assigned a unique global IP address for peering and protocol exchanges within and/or external to the transport network 10. As described in more detail below, the transport elements 52 may implement a distributed architecture in which local processors control each of the ports 34 and a centralized processor controls the network element 52.

In particular embodiments, the transport elements may be transport routers 60 interconnecting sets of subtending IP routers 14, transport Ethernet switches 62 interconnecting sets of subtending Ethernet switches 16, and transport Frame Relay switches 64 interconnecting sets of subtending Frame Relay switches 18. In addition, the transport element 52 may interconnect two ports transparently, in which case the port group 50 is user protocol independent.

FIGURE 2 illustrates details of the transport router 60 in accordance with one embodiment of the present invention. In this embodiment, the transport router 60 comprises a simple port group and acts as a single network element within a customer's autonomous network.

Referring to FIGURE 2, the transport router 60 includes geographically distributed ports 34 connected to external routers 14. The external ports 34 form a port group 50 with point-to-multipoint connectivity between the ports 34 as externally represented by the router 80. Accordingly, traffic from any one of the external routers 14 may be routed from an ingress port 34 directly to any number of the other external routers 14 by router 80.

The transport router 60 includes a router identifier to peer with the external routers 14 and participate in reservation and other protocol exchanges. In a particular embodiment, the transport router 60 peers with subtending routers 14 by using interior gateway protocols (IGP) such as OSPF, IS-IS, or RIP. The transport router 60 may peer using an exterior gateway protocol (EGP) or any other suitable protocol.

FIGURE 3 illustrates details of the IPT node 30 in accordance with one embodiment of the present invention. In this embodiment, the IPT node 30 comprises an add/drop multiplexer (ADM) with modular building blocks to support a scalable, pay-as-you-grow architecture. Accordingly, the transport network 10 owner may add functionality and incur cost based on customer demand. Functionality of the IPT node 30 and other components of the transport network 10 may be implemented by logic encoded in software and/or hardware media such as magnetic disks, application-specific integrated circuits (ASIC), field programmable gate arrays (FPGA) and the like.

Referring to FIGURE 3, the IPT node 30 includes one or more receiver-transceiver pairs (RTP) 100 and a processing system 102 interconnected by an internal Ethernet connection. As described in more detail below, each RTP 100 includes one or more internal interfaces 104 and one or more external interfaces 106. The internal interfaces are high-speed interfaces between the IPT nodes 30 while the external interfaces 106 are low-speed ports 34 accessible to external nodes. The internal and local interfaces 104 and 106 may each be implemented as one or more discrete cards.

Within the transport network 10, a set of internal interfaces 104 of the IPT nodes 30 are connected together between ports 34 of a port group 50 to form an FTS between the ports 34 and provide multipoint-to-multipoint and/or point-to-multipoint connectivity. In particular, a multiplexer of an internal interface 104 is connected to a demultiplexer of a next internal interface 104 in the FTS while a demultiplexer of the internal interface 104 is connected to a multiplexer of a previous internal interface 104 in the FTS. The FTSs are directionally-sensitive to preferentially route pass-through traffic over local ingress traffic. In this way, traffic for a transport element 52 is transported between an ingress and egress port on an FTS with minimal delay across the transport network 10.

The processing system 102 includes one or more central processing units (CPUs) 108. The CPUs 108 may each operate the IPT node 30 or a transport element 52. A CPU 108 operating the IPT node 30 includes an operating system and control functionality for the IPT node 30. A CPU 108 operating a transport element 52 includes control functionality for the distributed components of the transport element 52.

In one embodiment, a non-forwardable address space of the external network is used in the transport network

10 to route management and/or control traffic between processions as well as to and from the management station 20 or other external station. In a particular embodiment, the non-forwardable address space may be Internet Assigned Number Authority (IANA) reserved looped back address space. In this embodiment, the local interface 106 or other boundary interface is provided with NAT to map external IP addresses for messages generated outside the network to internal loop back addresses such that any, all or specified CPUs 108 and other components in the transport network 10 can be addressed in a suitable external address space.

The loop back address space may utilize a naming convention identifying the traffic as belonging to the loop back space and identifying the source and/or destination node and component of the node. In a particular embodiment, the naming convention comprises: 127, node identifier, port or CPU identifier. The 127 identifies the traffic as belonging to the IANA loop back address space. The node and the port or CPU identifiers may be a number or other unique identifier in the address space of the transport network 10.

In operation, the management station 20 or other external station generates a message for a CPU 108 or other addressable component of the transport network 10. The message includes a message data and external source and destination IP addresses. The message is forwarded using the IP addresses to a management ingress port or point of the transport network 10 corresponding to the destination IP address. At the ingress port, the external IP address are translated to the internal loop back traffic address space dynamically and/or using lookup tables. During translation, the external IP address is replaced with the loop back address identifier with the node and component also being transmitted based on included external identifiers of the node and component. In addition, the original source address is replaced with a management port or other suitable egress port address. The original source address is stored for translation of reply traffic.

The IPT nodes 30 are configured with a modified TCP/IP stack to route the loop back addressed traffic to an identified destination node for delivery to the destination port or CPU. Responses from a destination CPU 108 are routed to the management port, which is the egress port for response traffic, using the internal source address. At the management port, the destination port address is translated to the original source address for transmission in the external network and delivery to the management station 20. In this way, the internal topology is protected and many components are externally addressable using a reduced number of IP addresses which may be suitably scaled. Multiple loop back addresses may be assigned to an interface for communicating with multiple management stations 20. Processors in the transport network 10 may also use the loop back address space to communicated control messages. In this case, however, no translation is required.

FIGURE 4 illustrates details of the RTP 100 in accordance with one embodiment of the present invention. In this embodiment, the internal interface 104 is a high-speed interface that operates at substantially 10 Gb/s. The external interface 106 is a low-speed packet over SONET (POS) interface that operates at 2.5 Gb/s or below.

Referring to FIGURE 4, the internal interface 104 includes an optical receiver 110, a demultiplexer 112, a multiplexer 114, and an optical transmitter 116. The optical receiver is a 10 Gb/s receiver without SONET or package level knowledge. The optical receiver 110 performs the optical to electrical signal conversion. The optical receiver 110 may include an amplifier and may directly interface with a wave division multiplex (WDM) system.

The demultiplexer 112 drops local traffic and inter RTP traffic as well as buffers transit traffic. In a particular embodiment, the demultiplexer 112 has a set of 155 Mb/s connections to interface cards of the external interface 106. The demultiplexer 112 may also have 155 Mb/s connections to interface cards of other RTPs 100.

The multiplexer 114 collects local traffic from the interface cards of the external interface 106 and through traffic from the demultiplexer 112. The multiplexer 114 includes packet buffer, scheduler and insertion control functionality.

The optical transmitter 116 is a 10 Gb/s transmitter without SONET or package level knowledge. The optical transmitter 116 may include an optical amplifier. The optical transmitter 116 performs a conversion from an electrical signal to an optical signal and may interface directly with a WDM system.

The external interface 106 include a plurality of low-speed interface cards 120. The low-speed interface cards 120 send and receive traffic to and from the multiplexer 114 and demultiplexer 112, respectively. The low-speed interface cards 120 also provide connections between the FTSs.

The low-speed interface cards 120 are the main buffering point for ingress and egress traffic of the transport network 10. Packet level intelligence, including routing and protection mechanisms, are provided by the low-speed interface cards 120. If the transport network 10 uses an isolated addressing scheme, the low-speed interface cards 120 perform NAT functionality.

FIGURE 5 illustrates details of the processing system 102 in accordance with one embodiment of the present invention. In this embodiment, the transport network 10 includes an internal (IPTL1) layer and an external (IPTL2) layer. The processing system 102 provides a distributed architecture for the transport element 52. In particular, each port 34 of a transport element 52 is locally managed with control processing performed by a centralized processor.

Referring to FIGURE 5, the processing system 102 includes four CPUs 108 each configurable to operate the IPT node 30 or a transport element 52. The first CPU 140 manages the IPT node 30 and includes a simple network management protocol (SNMP) agent/internal network layer one (IPTL1) management information base (MIB) 142 for the IPT node 30. A common management information base (CMIB) 144 includes a model 146 of the transport network 10 and slave models 148 for transport elements having local ports. A database manager 150 manages the CMIB 144. An internal transport network layer one (IPTL1) architecture 152 includes an internal open shortest path first (IOSPF) instance 154 for discovery of the transport network 10. The IPTL1 architecture also includes control component subsystems 156.

The second CPU 160 is a master controller for a first transport element 52 of the transport network 10. The second CPU 160 includes an SNMP agent/external network MIB 162 for the first transport element 52. A CMIB 164 includes a master model 166 of the layer two (IPTL2) architecture for the first transport element 52. A database manager 168 manages the CMIB 166. The IPTL2 architecture 170 includes an OSPF instance 172 for discovery of the network connected to the first transport element 52. The IPTL2 architecture also includes control component subsystems 174.

The third CPU 180 is a master controller for a second transport element 52 of the transport network 10. The third CPU 180 includes an SNMP agent/external network MIB 182 for a second transport element 52. A CMIB 184 includes the master model 186 of the IPTL2 architecture for the second transport element 52. A database manager 188 manages the CMIB 184. The IPTL2 architecture 190 includes an OSPF instance 192 for discovery of the network connected to the second transport element 52. The IPTL2 architecture also includes control component subsystems 194.

The OSPF instances for each transport element discovers the topology for the element and generates the master model. The model is then distributed to the port controllers as slave models for point-to-multipoint connectivity within the port group of the transport element. The fourth CPU 198 is unassigned to a particular transport element 52 and may be idle or used to control lower layer functions.

In operation, layer one (IPTL1) learns the internal topology and does not exchange this information outside the transport network 10. The internal paths are learned using IPTL1 in order to route traffic between any two points within the network 10 regardless of the contents of the package. The traffic may be locally or externally generated. All IPT nodes 30 participate in IPTL1. Layer two (IPTL2) deals with the external topology for a transport router.

Each IPT node 30 is assigned a unique internal OSPF (IOSPF) router identifier. The transport network 10 runs IOSPF between the IPT nodes 30 to provide normal and protection paths between ingress points of the network. As a result, the transport network is modeled as a collection of routers interconnected by point-to-point links.

As described in more detail below, path label calculation (PLC) interacts with the IOSPF in order to learn the transport network 10 topology. Based on the learned topology, PLC determines the normal and protection paths. PLC also addresses overlapping paths. After PLC has learned the transport network topology, PLC signals IPTL2 to start running. When IPTL2 converges, OSPF is updated in the forwarding table for the corresponding transport element 52. PLC then populates the look-up table for the ports 34 of the transport element 52.

FIGURE 6 is a block diagram illustrating the distributed control architecture for transportation routers 60 in an exemplary network. The exemplary network includes a first IPT node 200, a second IPT node 202, a third IPT node 204, and a fourth IPT node 206.

The first IPT node 200 includes a first and second port for a first transport router, a first port for a second transport router, and a fourth and fifth port for a third transport router. The first CPU 210 includes control functionality for the first IPT node 200 as well as slave models of the first, second, and third transport routers for controlling the local ports. The second CPU 212 is a master controller for the first transport router.

The second IPT node 202 includes a third port of the third transport router and a third and fourth port of the second transport router. The first CPU 220 includes control functionality for the second IPT node 202 and slave models of the second and third transport routers for controlling the local ports. The second CPU 222 is a primary controller for the third transport router.

The third IPT node 204 includes the fourth port of the first transport router, a second port of the second transport router, and a first and second port of the third transport router. The first CPU 230 comprises control functionality for the third IPT node 204 and slave models of the first, second, and third transport routers for managing the local ports. The second CPU 232 includes a master controller for the second transport router.

The fourth IPT node 206 includes a third port of the first transport router and a fifth port of the second transport router. The first CPU 240 includes control functionality for the fourth IPT node 206 and slave models of the second transport routers for controlling the local ports. In this way, each IPT node and ports of the IPT node are locally managed. The distributed transport elements are managed by a centralized controller on any one of the IPT nodes.

FIGURE 7 illustrates the IPTL1 architecture 250 in accordance with one embodiment of the present invention. FIGURE 8 illustrates the IPTL2 architecture 260 in this embodiment in which the transport network 10 uses a transport label to efficiently transport traffic in the network 10. OSPF uses opaque link state advertisements (OLSAs) in order to discover the external network topology.

Referring to FIGURE 7, the functionality of the PLC 252 is based on whether the processor is managing an instance of IOSPF. An IPT node 30 will have only one instance of IOSPF, but each processor will have an instance of PLC 252. The PLC 252 instance associated with IOSPF builds a local configuration database (LDB) from IPTL1 and IPTL2 provision values, creates the OLSA entry from the configuration of IPTL1, tunnels the OLSA entry to IOSPF, retrieves the OLSA database from IOSPF upon IOSPF's notification of convergence, synchronizes the OLSA database with its PLC peers within an IPT node, signals IPTL2 to start by adding the transport router's port IP address, the multicast host, and transport router's IP address to the port prefix table and adding the CPU's label to the transport table of the port. The PLC 252 also receives the IPTL2 forwarding table (IP forwarding table), populates the prefixes, the transport labels and the destinations mapping tables for the ports of the IPTL2.

The PLC 252 receives fault signal from a fault manager which indicate the link failure identifier. In response to a link failure, the PLC 252 determines which label is effected by the link failure and marks the label as invalid in the transport label's table per port. If the link identifier is local, the OLSA conveys the failure and hands failure processing over to IOSPF.

The PLC 252 also translates an internal reservation protocol (RSVP) request on a normal path. The internal RSVP specifies the ingress and egress ports. The normal path includes a control path and a data path. A control path is a list of IPT nodes 30 to be traversed from a source to a destination. The data path is a list of high speed and slow speed links to be traversed between the source and the destination. If the internal RSVP succeeds in making a reservation on the normal path, it indicates to the PLC 252 the new QoS of the path. The PLC 252 updates the QoS of the normal transport label for the port 34. The same process occurs for the protection path. If the port 34 is not local to the PLC 252, the PLC 252 tunnels the information to the PLC 252 where the port resides to do the update. Further information regarding the internal reservation process is described in co-owned U.S. Patent Application entitled "System and Method for Opaque Application Object Transport", filed June 6, 2000.

The PLC 252 further supports a proprietary MIB for port lookup table and receives requests from MPLS. The requests include an IP destination prefix and an ingress port. The PLC 252 returns the pointers of the normal and protection transport labels and a next-hop IP address of the subtending router 14. The PLC 252 supports a device driver API to update the forwarding table in the port and supports a label translator to reach any point in the transport network 10.

The PLC 252 instance not associated with IOSPF builds a local configuration database (LDB) from IPTL1 and IPTL2 provisioned values, synchronizes the OLSA database with its IOSPF's PLC peers within an IPT node, signals IPTL2 to start by adding the transport router's port IP address, the multicast host, and transport router IP address to the port prefix table and adding the CPU's label to the transport table of the port, populates the prefixes, the transport labels, and the destinations mapping tables for the ports of the IPTL2.

The PLC 252 also receives fault signal from a fault manager which will indicate the link failure identifier. In this case the PLC 252 determines which label has been effected by the link failure and marks the label as invalid in the transport label's table per port.

The PLC 252 further translates an external IP address to IPTL2 to an egress port for external RSVP, receives signals from a PLC 252 associated with IOSPF to update the local port and receives an internal RSVP request on a normal path. As previously described, the internal RSVP will specify the ingress and egress ports. The normal path includes a control path and a data path. The control path is a list of IPT nodes 30 to be traversed from a source to a destination. The data path is a list of high speed links and low speed links to be traversed between the source and the destination. If the internal RSVP has succeeded in making reservation on the normal path, it indicates to the PLC 252 the new quality of service (QoS) of the path. The PLC 252 updates the QoS of the normal transport label for the port. The same process occurs for protection path. The PLC 252 also supports a device driver API to update forwarding table in ports and supports a label translator to reach any point in an transport network 10. To perform the necessary functions, IOSPF will include an API to permit the PLC 252 to pass the OLSA to the IOSPF, signal the PLC to retrieve OLSA database, modify OSPF link state database's structure to store and flood OLSA.

Referring to FIGURE 8, the IPTL2 architecture 260 comprises the topology for the transport router 60. The transport router manages the ports 34 in its ports group 50. The subtending routers 14 view the transport router 60 as a single router. The transport router 60 reacts to both external and internal changes in topology, which triggers updates between the subtending routers 14 and the transport router 60. Changes inside the transport network 10 that do not impact the states of the port 34 are not reported to the subtending routers 14.

As previously described, a master transport router instance resides in a single processor 262 within the transport network 10. Slave processors 264 resides on each transport node 30 including a port 34 for the transport router 60. Each processor 262 and 264 associated with the transport router 60 has a port group communication module 266.

A TCP connection is established between the transport routers instance and the ports instances. This connection is used to traffic control data between the transport router 60 and the subtending routers 14. The communication instance for the transport router 60 monitors the states of the transport routers ports 34 via the TCP connection with the ports instance, downloads a forwarding table upon notification from the routers OSPF, requests from the PLC 252 to translate a port 34 to a transport label, interacts with CMP 268 to send and receive packets, and tunnels the management's control packets to the transport routers ports 34. The ports communication instance establishes TCP connections with the transport router 60, tunnels all control packets to the transport router 60, request from the PLC 252 to translate a port 34 to a transport label, receives a forwarding table from the transport router 60 and downloads a forwarding table to the PLC 252.

FIGURE 9 illustrates a method for provisioning transport elements 52 in the transport network 10 in accordance with one embodiment of the present invention. The method begins at step 350 in which connections are provisioned between the IPT nodes 30. The connections define the FTSs within the transport network 10. At step 352, addresses for each transport elements 52 are defined within the address space for the IPT network 10.

Proceeding to step 354, the internal topology of the transport network is discovered. At step 356, transport elements 52 are defined within the transport network 10. The transport elements 52 each comprise a port group 50 and may be a transport router, transport Ethernet switch, or transport Frame Relay switch. At step 358, topology of the transport elements 52 and connected external nodes are discovered.

Next, at step 360, the transport elements 52 each peer with the subtending routers 14 or other external nodes. At step 362, the transport elements 52 generate routing tables for receiving and transmitting packets to and from the external network and within the transport network 10. In this way, the transport elements 52 are freely defined within the transport network 10 to match the topology of the network 10 to needs of customers.

FIGURE 10 illustrates a method for defining a transport element 52 in the transport network 10 in accordance with one embodiment of the present invention. The method begins at step 400 in which a master, or primary processor for the transport element 52 is assigned within the transport network 10. As previously described, the master processor controls the transport element 52 directly and through slave processors local to each of the ports 34. Next, at step 402, ports 34 are identified and assigned to the transport element 52.

Proceeding to step 404, a local processor is assigned or otherwise provided for each port 34 of the transport element 52. In one embodiment, the local processor by default is a master processor for each corresponding IPT node 30. At step 406, an identifier is assigned to the transport element 52 to allow the transport element 52 to participate in protocol exchanges and otherwise appear as a single element to external nodes.

FIGURE 11 illustrates a method for generating routing tables for a transport element 52 in accordance with one embodiment of the present invention. The method begins at step 450 in which a routing information base (RIB) is generated by a master processor for a transport element 52. The RIB is generated based on the IPTL1 and IPTL2 architectures.

At step 452, the RIB is distributed to each port 34 of the transport element 52. At step 454, a forwarding information base (FIB) is generated at each port 34 based on the RIB. The ports 34 use the RIB to process traffic received from the transport network 10 or the external network 12. Step 454 leads to the end of the process by which routing information is centrally generated and distributed for the transport element 52.

FIGURE 12 illustrates a method for processing through traffic in a transport element 52 in accordance with one embodiment of the present invention. The method begins at step 500 in which an IP packet is received at an ingress port 34 of a transport element 52. At step 502, a transport label is generated based on the IP address using the FIB for the transport element 52.

Proceeding to step 504, the transport label is added to the IP packet to generate an internal packet. At step 506, the internal packet is transported to an egress port 34 of the transport element 52 on high-speed links based on the transport label.

Next, at step 508, the transport label is removed from the IP packet at the egress port 34. At step 510, the IP packet is transmitted to an external destination element. Step 510 leads to the end of the process by which IP packets are transmitted across the transport network 10 on high speed links using transport labels overhead.

FIGURE 13 illustrates a method for routing messages in the transport network using a unique internal address space and for translating between the internal address space and an external IP address space in accordance with one embodiment of the present invention. In this embodiment, the IANA reserve loop back address space is used to address and route messages within the transport network 10. This may allow the internal topology to be protected from the external network, prevent conflicts between address spaces of the transport network and the external network or networks, and allow a single or reduced set of IP addresses to be used by a management or other external station to address components of the transport network 10.

Referring to FIGURE 13, the method begins at step 550 in which a management station 20 generates message data for an element of the transport network 10. The element may be a CPU 108, port, RTP 100 or other suitable component of an IPT node 30 operable to be remotely managed. At step 552, the message data is addressed with external IP addresses for forwarding in the external network. The IP addresses include the source address of the management station and the destination address of a management port of the transport network 10. The destination address also includes an external identifier of the transport node and component. At step 554, the message is routed to the management port based on the external IP addresses.

Proceeding to step 556, at the boundary interface of the transport network, the IP source address is stored for addressing of reply traffic. At step 558, the IP addresses are translated to internal loop back addresses. In a particular embodiment, the destination address may be translated to a 127 loop back address space and internal node and component identifiers using a look up table. The source address is translated to the management port of the transport network for reply traffic. The message is routed to the destination element in the transport network 10 based on the internal loop back address.

Proceeding to step 562, the network element processes the message and generates a response. At step 564, the response is addressed with the internal loop back addresses of the component and the management port. At step 566, the response is routed to management port based on the internal loop back address.

Next, at step 568, the internal addresses are translated to the external IP address. In one embodiment, the destination loop back address of the management port is replaced with the original IP source address. At step 570, the response is routed to the base station 20 that originated the message based on external IP addresses. It will be understood that message may be otherwise addressed and routed to and within the transport network 10 without departing from the scope of the present invention.

Although the present invention has been described with several embodiments, various changes and modifications may be suggested to one skilled in the art. It is intended that the present invention encompass such changes and modifications as fall within the scope of the appended claims.